

# **Absolvování individuální odborné praxe**

## **Individual Professional Practice in the Company**

## Zadání bakalářské práce

Student: **Martin Kriváček**  
Studijní program: B2647 Informační a komunikační technologie  
Studijní obor: 2601R013 Telekomunikační technika  
Téma: **Absolvování individuální odborné praxe**  
**Individual Professional Practice in the Company**

Zásady pro vypracování:

1. Student vykoná individuální praxi ve firmě: Siemens Enterprise Communications s.r.o.
2. Struktura závěrečné zprávy:
  - a. Popis odborného zaměření firmy, u které student vykonal odbornou praxi a popis pracovního zařazení studenta
  - b. Seznam úkolů zadaných studentovi v průběhu odborné praxe s vyjádřením jejich časové náročnosti
  - c. Zvolený postup řešení zadaných úkolů
  - d. Teoretické a praktické znalosti a dovednosti získané v průběhu studia uplatněné studentem v průběhu odborné praxe
  - e. Znalosti či dovednosti scházející studentovi v průběhu odborné praxe
  - f. Dosažené výsledky v průběhu odborné praxe a její celkové zhodnocení

Seznam doporučené odborné literatury:

Podle pokynů konzultanta, který vedl odbornou praxi


Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

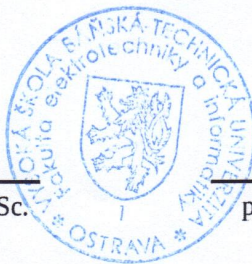
Vedoucí bakalářské práce: **Ing. Zdeňka Chmelíková, Ph.D.**

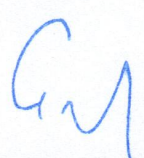
Konzultant bakalářské práce: Ing. Mitko Mitěv

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

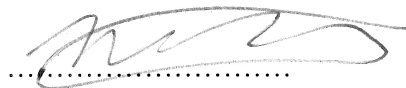
  
prof. RNDr. Vladimír Vašínek, CSc.  
vedoucí katedry



  
prof. RNDr. Václav Snášel, CSc.  
děkan fakulty

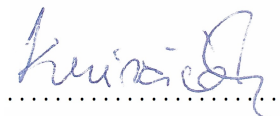
Súhlasím so zverejnením tejto bakalárskej práce podľa požiadavku čl.26, ods.9 *Studijného a zkušebního řádu pro studium v bakalářských programech VŠB-TU Ostrava*.

V Brně 29.4.2013

A handwritten signature in black ink, consisting of stylized, flowing letters, positioned above a dotted line.

Prehlasujem, že som túto bakalársku prácu vypracoval samostatne. Uviedol som všetky literárne pramene a publikácie, z ktorých som čerpal.

V Ostrave 29.4.2013

A handwritten signature in blue ink, consisting of stylized, flowing letters, positioned above a dotted line.

Rád by som na tomto mieste poďakoval za pomoc pri vytváraní tejto práce a taktiež za moje vedenie vo firme počas celej bakalárskej praxe pánovi Ing.Mitkovi Mitěvovy.

## **Abstrakt**

Cieľom tejto bakalárskej práce je popísať priebeh mojej praxe v spoločnosti Siemens Enterprise Communication s.r.o., globálneho poskytovateľa komunikačných služieb. Počas praxe som nahrádzal pôvodné grafické rozhranie aplikácie OSCAR-ITT novým rozhraním za použitia platformy Java FX.

V prvej časti predstavím spoločnosť a moje pracovné zameranie, v druhej časti sa budem venovať priamo úlohám ktoré mi boli zadané a ich riešeniam. Posledná časť bude obsahovať subjektívne zhodnotenie dosiahnutých výsledkov a skúsenosti získaných počas praxe.

**Kľúčová slova:** Odborná prax, Siemens EN, Java FX, GUI, OSCAR-ITT

## **Abstract**

The thesis aims on description of my practice in Siemens Enterprise Communication Ltd., the global provider of communications services. I was working at replacing the original graphical user interface OSCAR-ITT, by the new interface using Java FX during my practice.

In first part describes company and my assignment, at second part describes project and task i was resolving. The last part concludes achieved results and experiences I have gained.

**Keywords:** Professional practice, Siemens EN, Java FX, GUI, OSCAR-ITT

## Seznam použitých zkratek a symbolů

API	– Application Programming Interface
AWT	– Abstract Window Toolkit
B2BUA	– Back To Back User Agent
CSV	– Comma-Separated Values
DNA	– DeoxyriboNucleic Acid
GUI	– Graphic User Interface
HTTP	– Hypertext Transfer Protocol
JVM	– Java Virtual Machine
OS	– Operating System
PBX	– Private Branch Exchange
IDE	– Integrated Development Environment
MVC	– Model View Controller
SDP	– Session Description Protocol
SDK	– Software Development Kit
SIP	– Session Initiation Protocol
SMTP	– Simple Mail Transfer Protocol
UI	– User Interface
UA	– User Agent
XML	– eXtensible Markup Language

# Obsah

<b>1</b>	<b>Úvod</b>	<b>9</b>
<b>2</b>	<b>Predstavenie spoločnosti</b>	<b>10</b>
2.1	Pracovné zaradenie . . . . .	10
<b>3</b>	<b>Predstavenie projektu</b>	<b>11</b>
3.1	Oscar-ITT . . . . .	11
3.2	SIPp . . . . .	11
3.3	SIP protokol . . . . .	11
<b>4</b>	<b>Riešené úlohy</b>	<b>13</b>
4.1	Prehľad použitých technológií . . . . .	13
4.2	Prehľad grafických knižníc a tvorba nového GUI . . . . .	15
4.3	Nahradenie pôvodného GUI v aplikácii OSCAR-ITT . . . . .	15
4.4	Implementácia základných funkcií aplikácie - Hlavné okno aplikácie . . .	16
4.5	Implementácia základných funkcií aplikácie - Dialógové okná . . . . .	18
4.6	Implementácia základných funkcií aplikácie - Spúšťanie testov . . . . .	19
4.7	Ostatné dialógové okná . . . . .	20
4.8	Hlavné problémy . . . . .	20
<b>5</b>	<b>Záver</b>	<b>22</b>
5.1	Uplatnené znalosti získané počas štúdia . . . . .	22
5.2	Chýbajúce znalosti . . . . .	22
5.3	Zhodnotenie praxe . . . . .	22
<b>6</b>	<b>Literatúra</b>	<b>23</b>

## Seznam obrázků

1	Výzor pôvodného hlavného okna aplikácie . . . . .	15
2	Môj návrh hlavného okna aplikácie . . . . .	16
3	Ukážka stromu a kontextového menu . . . . .	17
4	Grafický panel . . . . .	18
5	Spustený test . . . . .	20



## 1 Úvod

Rozhodol som sa absolvovať bakalársku prácu formou bakalárskej praxe, pretože verím že bude mať pre mňa väčší prínos ako riešenie zadanej témy. Vidím to ako spôsob vďaka ktorému získam praktické skúsenosti, ktoré sú v tejto dobe tak veľmi žiadané. Preto som sa rozhodol požiadať o možnosť praxe práve vo firme Siemens Enterprise Communication ktorá ma bohaté zázemie v oblasti telekomunikačnej techniky a takto nadobudnuté skúsenosti sa mi budú v budúcnosti určite užitočné.

V tejto práci budem popisovať priebeh mojej praxe vo firme Siemens Enterprise Communication s.r.o. V prvej časti predstavím spoločnosť a moje pracovné zameranie, v druhej časti sa budem venovať priamo úlohám ktoré mi boli zadane a ich riešeniam. Posledná časť bude obsahovať subjektívne zhodnotenie dosiahnutých výsledkov a skúseností získaných počas praxe.

## 2 Predstavenie spoločnosti

Siemens Enterprise Communications, je globálne integrovaný poskytovateľ komunikačných služieb, ktorý synchronizuje, nasadzuje a spravuje technológie, ako sú hlasové a video služby, možnosti spolupráce, mobility, kontaktné centrum, a sieťovú infraštruktúru. Privádza tieto komunikačné technológie priamo do cesty ktorou podniky fungujú. Výsledkom je transformácia spôsobu , ako podnik komunikuje a spolupracuje - ktorá zlepšuje kolektívne úsilie a výrazne zlepšuje výkonnosť podniku.

Zrodila sa z technického DNA spoločnosti Siemens, bola vybudovaná na tomto dedičstve - spoľahlivosti produktov, inovácie, otvorených štandardov a bezpečnosti poskytovania integrovaných komunikačných riešení pre viac ako 75% z Global 500. Siemens Enterprise Communications je spoločný podnik Gores Group a Siemens AG.

Fakty o spoločnosti Siemens Enterprise Communications:

- Viac ako jeden milión podnikovým zákazníkov v takmer každom priemyselnom odvetví
- Približne 150.000 volaní za minútu sú inicializovaných pomocou technológii spoločnosti Siemens
- Viac ako tri milióny globálne spravovaných hlasových a dátových portov
- Dostupnosť služieb v 90 krajinách priamo alebo prostredníctvom partnerstiev
- Víťazne ocenený globálny partnerský program, Go Forward!, s 500 akreditovanými partnermi po celom svete

[1]

### 2.1 Pracovné zaradenie

Po prijímacom pohovore mi bol priradený samostatný projekt. Siemens EN má testovaciu aplikáciu OSCAR-ITT napísanú v Jave. Fron-end aplikácia využíva Swing API. Toto API je v súčasnej dobe už nemoderné a bolo by dobré ho nahradiť modernejším a viac užívateľsky prívetivejším rozhraním. Nové rozhranie by malo zároveň poskytovať moderný vzhľad odrážajúci novšie UI ako sú Windows 7/8, Android, iOS.

## 3 Predstavenie projektu

### 3.1 Oscar-ITT

Je aplikácia ktorá sa zaoberá testovaním komunikácie medzi členmi siete pracujúcimi pomocou protokolu SIP. Aplikačná logika je napísaná v jazyku Java. Pôvodné grafické rozhranie bolo postavené na grafických knižniciach Swing a AWT. Aplikácia využíva na testovanie open source nástroj SIPp.

### 3.2 SIPp

SIPp je open source testovací nástroj / generátor sieťového zaťaženia pre SIP protokol. Zahŕňa niekoľko základných SipStone scenárov, umožňuje načítanie vlastných XML scenárov popisujúce od jednoduchých až po komplexné Výmeny Správ (Call Flows). Dokáže dynamicky zobrazovať štatistiky o bežiacich testoch, pravidelné zhromažďovanie CSV štatistík. Je používaný na testovanie množstva reálnych SIP zariadení ako napríklad: SIP proxy, B2BUAs, SIP média servery, SIP / x brány, SIP PBX.. A taktiež veľmi užitočné emulovanie tisícok užívateľských agentov používajúcich váš SIP systém. [7]

### 3.3 SIP protokol

SIP je signalizačný protokol určený na vytváranie a ukončovanie multimediálnych spojení (telefónny hovor, videohovor) cez Internet. Ďalšie možné použitia sú hlasové a videokonferencie, rýchle správy, streamovanie multimediálnych dát. Spojenie môže predstavovať akýkoľvek multimediálny prenos, v praxi sa však najčastejšie SIP používa na telefonovanie po IP sieti.

SIP je textový protokol, využíva prvky dvoch veľmi často používaných internetových protokolov a to protokolu HTTP, ktorý slúži na prehliadanie internetových stránok a protokolu SMTP, slúžiaceho na prenos elektronickej pošty. Správy samotného protokolu sú vždy požiadavky (requests) alebo odpovede (responses).

Protokol zabezpečuje možnosť vytvorenia multimediálneho dialógu medzi dvomi koncovými bodmi. Za hlavné signalizačné úlohy sa považujú:

- registrácia účastníkov
- vyhľadanie volaného účastníka
- zisťovanie jeho vôle k akceptovaniu spojenia
- výmena informácií o parametroch spojenia (SDP protokol, aké kodeky sa použijú G.711u-law, G.711a-law, g723 ...), aké rozšírenia protokolu môžu byť použité (Supported hlavička, 100rel)
- modifikácia existujúcich spojení (reINVITE, INVITE s existujúcim Dialog ID)
- ukončenie existujúcich spojení.

Takisto je v protokole implementovaná možnosť vyžiadania a doručenia údajov o prítomnosti jednotlivých účastníkov (presence napríklad, či daný účastník je pripojený /odpojený, alebo kde sa nachádza), rovnako ako možnosť posielania rýchlych správ (instant messages). SIP je nezávislý na type multimediálnych dát, ktoré sa prenášajú. V priebehu spojenia slúži skutočne iba na prenos informácií o účastníkoch, ale nijako nie je zviazaný s konkrétnym typom multimediálnych dát. V jednej relácii je možné prenášať vzorky zvuku, vzorky obrazu ale aj čistý text, atď. Na prenos samotných multimediálnych dát sa najčastejšie používa Real-time Transport Protocol (RTP) [6]

## 4 Riešené úlohy

Úlohy ktoré som riešil by som rozdelil do dvoch etáp:

- Preskúmať najnovšie knižnice pre užívateľské rozhrania, napísané pre Javu. Rozhranie musí byť buď open source alebo byť súčasťou Java SDK. Naprogramovať jednoduchú aplikáciu simulujúcu súčasné rozhranie aplikácie OSCAR-ITT za použitia vybraných knižníc. Na základe týchto prototypov by sa mali vyhodnotiť výhody použitia týchto knižníc a zároveň ukázať na príslušné rizika. Na základe tohto hodnotenia by sa vybrala vhodná knižnica pre ďalšiu fázu.
- Nahradenia Swingu vybranou knižnicou priamo v aplikácii OSCAR-ITT. V rámci nahradenia navrhnúť i lepšiu štruktúru aplikácie, tak aby lepšie odpovedala modelu MVC (model-view-controller).

### 4.1 Prehľad použitých technológií

Celý projekt som riešil hlavne vo vývojom nástroji Netbeans IDE v ktorom som implementoval hlavne logiku aplikácie. Na vytváranie grafického rozhrania som použil nástroj JavaFX Scene Builder. A celá aplikácia bola napísaná v programovacom jazyku Java.

#### 4.1.1 NetBeans IDE

NetBeans je integrované vývojové prostredie (IDE) pre vývoj aplikácií predovšetkým pomocou programacieho jazyka Java, ale aj s iných jazykov, najmä PHP, C / C++ a HTML5. Taktiež to je aplikačný framework pre Java desktopové aplikácie a iné.

NetBeans IDE je napísaný v Jave a môže bežať ako na Windows tak na OS X, Linux, Solaris a ďalších platformách, ktoré podporujú kompatibilné JVM.

Platforma NetBeans umožňuje vytváranie aplikácií, zo súboru modulárnych softvérových súčastí tzv. modulov. Aplikácie založené na platforme NetBeans (NetBeans IDE vrátane jeho samotného), môžu byť rozšírené vývojármi tretích strán. [2]

#### 4.1.2 JavaFX Scene Builder

JavaFX Scene Builder je vizuálny nástroj, ktorý používateľom umožňuje rýchlo navrhnuť JavaFX užívateľské rozhranie aplikácie, bez potreby programovania. Užívateľia môžu pretiahnutím UI komponenty na pracovnú plochu a tak túto komponentu použiť, meniť jej vlastnosti, použiť štýly a FXML kód, ktorý sa vytvára automaticky je generovaný na pozadí. Výsledkom je FXML súbor, ktorý potom môžete kombinovať s Java projektom prepojením používateľského rozhrania s aplikačnou logikou. [3]

### 4.1.3 Java

Java je objektovo orientovaný programovací jazyk, ktorý vyvinula firma Sun Microsystems a predstavila 23. mája 1995. Java je jedným z najpoužívanějších programovacích jazykov na svete. Podľa Tiobe indexu je Java druhý najpopulárnejší programovací jazyk v tesnom závесе za jazykom C. Vďaka svojej prenositeľnosti je používaný pre programy, ktoré majú pracovať na rôznych systémoch od čipových kariet (platforma JavaCard), cez mobilné telefóny a rôzne zabudované zariadenia (platforma Java ME), aplikácie pre desktopové počítače (platforma Java SE) až po rozsiahle distribuované systémy pracujúce na rade spolupracujúcich počítačov rozprestrené po celom svete (platforma Java EE). Tieto technológie sa ako celok nazývajú platforma Java.

8. mája 2007 Sun uvoľnil zdrojové kódy Javy (cca 2,5 miliónu riadkov kódu) a Java je ďalej vyvíjaná ako open source.

Existuje niekoľko teórií o pôvode mena tohoto jazyka, jedna z nich hovorí o inšpirácii u slangového označenia kávy. [4]

### 4.1.4 Java FX

Java FX je softvérová platforma postavená na báze platformy Java z dielne spoločnosti Sun Microsystems. Slúži pre vývoj tzv. RIA aplikácií (Rich Internet Applications), v preklade bohatých internetových aplikácií. Jak už z názvu vyplýva, jedná sa predovšetkým o prvky zaisťujúce interaktivitu webových stránok, teda stránok, ktoré s užívateľom, oproti tým statickým, nejakým spôsobom komunikujú. V prípade Java FX webových aplikácií, ktoré sa formou pluginu spúšťajú z prehliadača, platforma taktiež umožňuje si aplikáciu stiahnuť a využívať ju ako klasický desktopový program a to i v offline móde. JavaFX je cielená na vývojárov programov túžiacich po rýchlom a jednoduchom vývoji pre desktopy, internetové prehliadače, ale i rýchlo sa rozvíjajúcu sa mobilnú oblasť. V budúcnosti sa očakáva rozšírenie platformy i pre ďalšie zariadenia, ako sú herné konzole alebo televízne set-top boxy. [5]

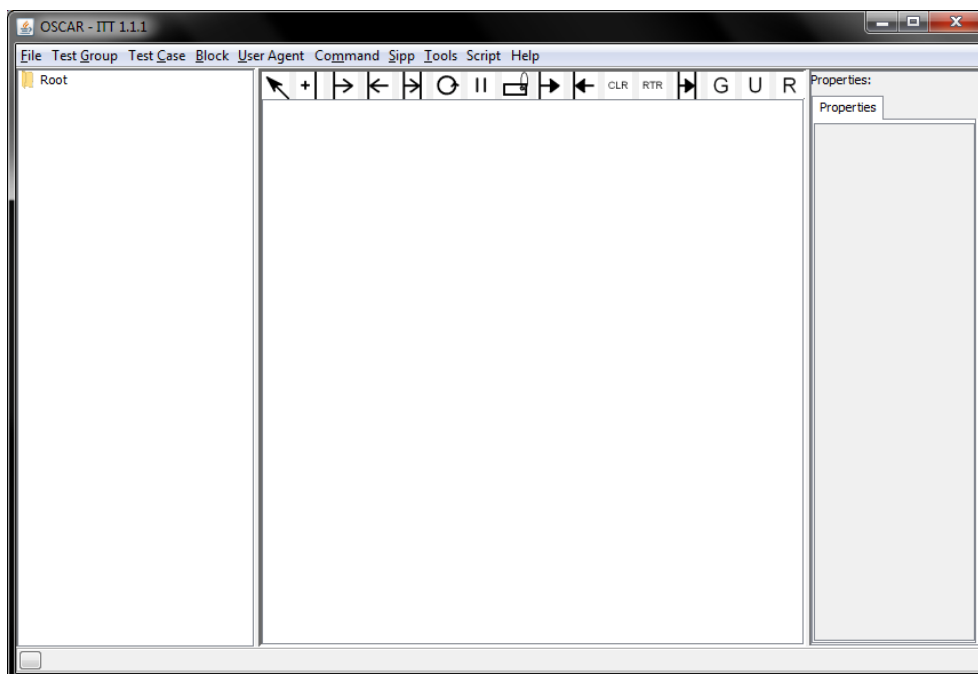
## 4.2 Prehľad grafických knižníc a tvorba nového GUI

### 4.2.1 Výber grafickej knižnice pre tvorbu GUI

Pri výbere grafickej knižnice som zvažoval viac možností, ako najlepšia možnosť dopadla platforma Java FX ktorá má bohaté možnosti tvorby komplexných grafických rozhraní.

### 4.2.2 Návrh Java FX GUI

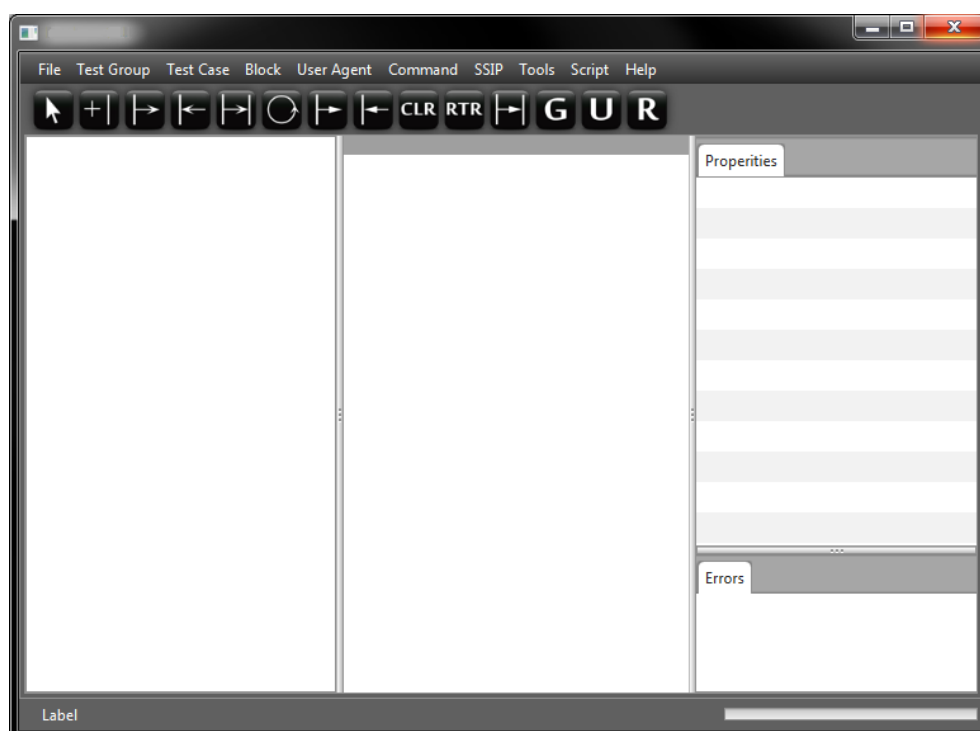
Pre návrh grafického rozhrania som použil Java FX Scene Builder, čo je program vyvinutý priamo pre platformu Java FX, ktorý umožňuje jednoducho pomocou „drag and drop“ vytvoriť okno aplikácie a následne ho uložiť vo formáte „xml“. Pre nastavenie detailov ako sú farby jednotlivých plôch som použil kaskádové štýly. Pri návrhu som sa riadil výzorom pôvodného GUI a vlastným uvážením.



Obrázek 1: Výzor pôvodného hlavného okna aplikácie

## 4.3 Nahradenie pôvodného GUI v aplikácii OSCAR-ITT

Po predvedení môjho návrhu nového GUI, bol tento návrh schválený a tým sme sa dostali do druhej fázy projektu. V tejto fáze som mal použiť môj návrh priamo v aplikácii OSCAR-ITT. Po podpísaní zmluvy o mlčanlivosti mi boli doručené zdrojové kódy. Prvých pár dní som strávil prezeraním a študovaním týchto kódov aby som vedel, kde začať s úpravami na prechod na nové GUI. Pôvodná aplikácia bola písaná tak, aby zodpovedala modelu



Obrázek 2: Môj návrh hlavného okna aplikácie

MVC. Ale aj napriek tomu obsahovala množstvo tried, ktoré patrili k GUI ale taktiež zasahovali do aplikačnej vrstvy a naopak.

Tuto implementáciu by som rozdelil na dve časti a to na časť prvú, v ktorej sa má obnoviť základná funkčnosť aplikácie a na časť druhú, v ktorej budem mať za úlohu previesť ďalšie vedľajšie a podporné funkcie.

#### 4.4 Implementácia základných funkcií aplikácie - Hlavné okno aplikácie

Zaistením základných funkcií aplikácie sa rozumie možnosť vytvárať testy, načítať uložené testy a následne ich spúšťať. K zaisteniu týchto funkcií bolo treba implementovať hlavné okno aplikácie.

##### 4.4.1 Hlavný strom

Jedná sa o dôležitý prvok hlavného okna v ktorom sa zobrazuje aktuálne otvorená štruktúra testov či testových skupín. Užívateľ môže za pomoci tohto stromu spravovať tieto testy, či už formou pridávania alebo odoberania jednotlivých súčastí, ako aj vytváraním nových testov alebo otváraním uložených. Taktiež môže prevádzať rôzne akcie dostupné z kontextového menu stromu. V hlavnom okne sa tento strom nachádza na ľavej strane.





Obrázek 3: Ukážka stromu a kontextového menu

Na tento účel bol v pôvodnej aplikácii použitý JTree z knižnice Swing, ja som použil komponentu TreeView, ktorá zobrazuje kolekciu objektov typu TreeItem. Strom obsahuje taktiež kontextové menu, v ktorom sa nachádzajú funkcie a príkazy v závislosti na aktuálne vybranom prvku stromu.

#### 4.4.2 Tabuľka vlastností

Pri výbere niektorého uzlu v hlavnom strome, sa v tabuľke zobrazia aktuálna vlastnosti a nastavenia prvku, ktorý sme vybrali či už v strome alebo v grafickej časti hlavného okna. Pri výbere testu sa taktiež zobrazí v pravom dolnom rohu okna tabuľka z nastaveniami pre program SIPp a konfiguračnými dátami pre jednotlivých UA.

Funkciu tabuľky zastupuje trieda FXPropTable, ktorá dedí z triedy TableView, ak vybraný prvok stromu obsahuje konfiguračné dáta pre program SIPp, zobrazí sa taktiež panel na ich zobrazenie a úpravu, reprezentovaný triedou FXSippPropPanel.

#### 4.4.3 Informačné karty

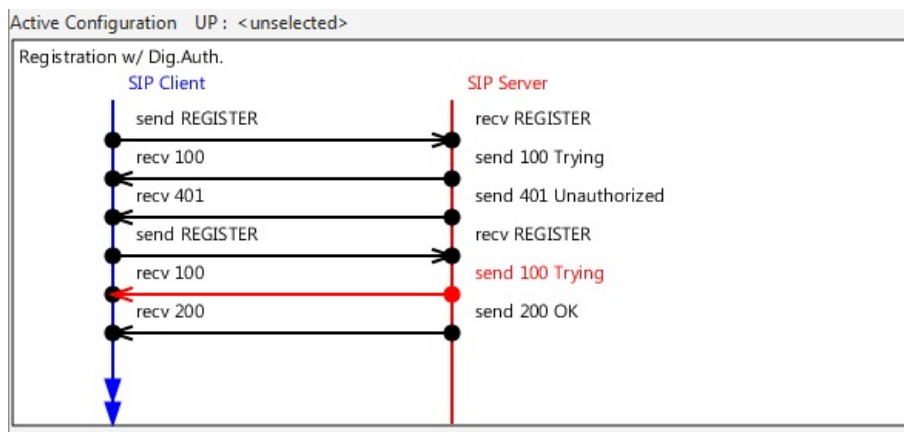
Slúžia na zobrazenie scriptov pre jednotlivých SIP klientov v teste a taktiež na zobrazenie výsledkov spustených testov. Nachádzajú sa na pravom paneli.

Pri požiadavke na zobrazenie scriptu sa zobrazí nová karta pre každú konfiguráciu, ktorá je priradená danému SIP klientovi. Na zobrazenie je použitá štandardná TextArea. Tento script je následne pri požiadavke na spustenie testu použitý na simulovanie SIP klienta.

Pri spustení testu sa zobrazí karta, ktorá obsahuje okno s priebehom testu a okno s chybovými hláškami. Táto karta je reprezentovaná triedou FXPanelSippOutput.

#### 4.4.4 Grafický panel

Nachádza sa v prostriedku hlavného okna, slúži na zobrazenie a úpravu blokov, z ktorých každý obsahuje jednu výmenu správ medzi SIP klientmi. V tomto okne je možné tieto výmeny správ upravovať alebo vytvárať úplne nové za pomoci jednoduchej grafiky.



Obrázek 4: Grafický panel

Funkčnosť kresliaceho plátna zastupuje trieda DrawPanelFX.

Súčasťou grafického panela je aj panel tlačidiel, ktorý je umiestnený nad grafickým panelom. Tlačidlá zastupujú funkčnosti kurzora pre konkrétny typ člena.

#### 4.4.5 Hlavné menu

Súčasťou okna je taktiež aj hlavné menu aplikácie v ktorom sa nachádzajú funkcie ako z kontextového menu hlavného stroma tak aj ďalšie akcie ktoré spúšťajú rôzne dialógové okná.

Všetky metódy ktoré sú pridelené k tlačidlám menu k funkcii OnAction sa nachádzajú v triede OscarControler.java.

### 4.5 Implementácia základných funkcií aplikácie - Dialógové okná

Pre základnú funkčnosť aplikácie sú potrebné taktiež niektoré dialógové okná. Tieto okná umožňujú meniť hlavné nastavenia aplikácie dôležité pre beh programu, ako aj možnosť otvárať a ukladať testy.

#### 4.5.1 Open Test Case Group/Test Case/Block

Funkciou tohto dialógového okna je možnosť otvárať dopredu pripravené testy alebo skupiny testov.

Pri otvorení dialógového okna sa v prípade testov zobrazuje v prvej polovici okna list testov, ktoré sa nachádzajú v adresári určenom pre testy, po jeho vybraní sa zobrazí

náhľad v druhej polovici okna. V prípade skupín testov sa popri liste zobrazuje aj záložka zo stromovou štruktúrou týchto skupín. Pri výbere testu sa zobrazuje v druhej polovici okna iba náhľad obsahu testu reprezentovaný triedou DrawPanelFX. Zatiaľ čo pri výbere skupiny testu sa zobrazí v tejto časti okna aj list v ktorom si vyberieme položku, ktorú chceme zobrazit'. Celá táto komponenta je reprezentovaná triedou FXTestGroupSplitPane.

Toto okno obsahuje taktiež aj možnosť vyhľadať položku podľa názvu. Pre tento účel sa na spodnej strane okna nachádza textové pole ktoré reaguje na zmenu obsahu a následne vyberte v lište prvú odpovedajúcu položku obsahujúcu zadaný reťazec.

## 4.5.2 Options

Dialógové okno options slúži k nastaveniu rôznych konfiguračných dát potrebných pre správne fungovanie aplikácie, nachádzajú sa tu od možnosti nastavenia umiestnenia repozitárov, cez nastavenia zobrazovania popiskou v blokoch až po možnosti logovania štatistík do súborov. Toto okno je reprezentované triedou FXDialogOption odvodené od triedy Stage.

## 4.5.3 UserProfiles

Toto okno slúži na spravovanie užívateľských profilov, t.j. vytváranie, úpravu a výber aktívneho profilu, ktorý sa používa pri testoch. Celé okno je reprezentované triedou FXDialogUserProfiles. Obsahuje list so zoznamom vytvorených profilov a tabuľky na úpravy.

## 4.5.4 SIP message parameters

Jedná sa o okno, ktoré umožňuje nahrádzať v konfiguračnom súbore kľúčové slová, ako napríklad IP adresu alebo užívateľské meno a heslo. Čím sa zaručuje prispôsobiteľnosť jednotlivých konfiguračných dát pre ich opätovné využívanie. Okno je implementované za pomoci triedy FXCustomPropertiesEdit a tabuľka pre zmenu týchto parametrov je reprezentovaná triedou FXTableProperties.

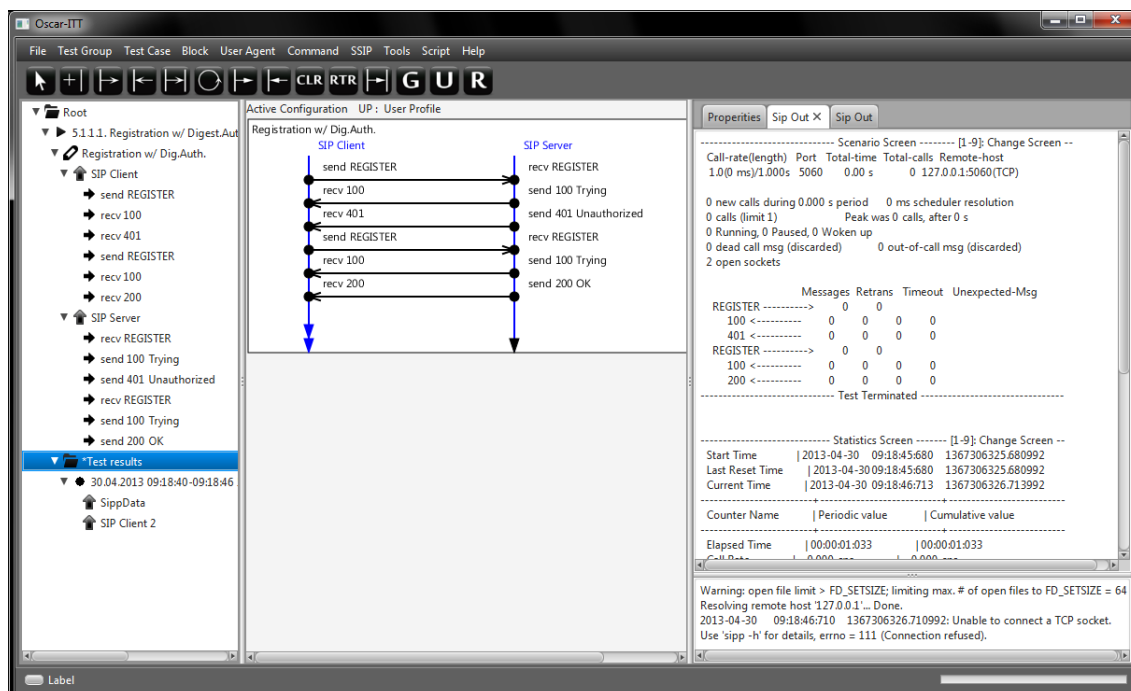
## 4.6 Implementácia základných funkcií aplikácie - Spúšťanie testov

Ako bolo vyššie spomenuté, cieľom obnovenia základnej funkcie bola možnosť spúšťať testy.

Po spustení testu sa na pozadí spustí nástroj SIPp z parametrami a dátami, ktoré sme si nastavili vopred v aplikácii. Taktiež sa vytvorí karta v pravej časti programu. V tejto karte sa sa v hornej časti zobrazuje štandardný výstup aplikácie, kde sú zobrazené rôzne štatistiky o priebehu testu. V spodnej časti sa zobrazuje chybový výstup.

Po skončení testu sa v strome vytvorí pri spustení teste nový priečinok, ktorý obsahuje výsledky tohto testu.

Výstupy z nástroja SIPp sú riešené jednoducho pomocou komponenty TextArea a pre každý test sa vytvorí nová karta obsahujúca tento výstup.



Obrázek 5: Spustený test

## 4.7 Ostatné dialógové okná

Pre úplnú funkčnosť aplikácie je treba implementovať ešte niekoľko podporných dialógových okien, ktoré sa starajú o správu repozitárov.

### 4.7.1 Clean Up Repository

Toto okno slúži na čistenie repozitárov od súčastí, ktoré už nebudeme využívať. Pokiaľ konkrétne vybraná súčasť, povedzme test nieje obsiahnutý v inej skupine testov, je možné tento test fyzicky odobrať z repozitárov.

### 4.7.2 Import/Export Repository

Slúži na vkladanie repozitárov do hlavného repozitára aplikácie alebo naopak na exportovanie celého repozitára na iné umiestnenie.

Funkčnosť okna je zaistená triedou FXReposMaintenance.

## 4.8 Hlavné problémy

Hlavným problémom, s ktorým som sa potýkal pri práci na tomto projekte bolo hlavne menšie množstvo informácií k technológii Java FX. Keďže táto technológia je pomerne nová takže je pomerne ťažké nájsť riešenie niektorých problémov s ktorými som sa stretol.

Preto som sa registroval vo fóre *javafx-jira.kenai.com*, kde sa mi podarilo pomocou informácií z tohoto fóra vyriešiť niektoré problémy. Bežnejšie problémy ktoré sa vyskytli som našiel aj na fóre *Stackoverflow.com*.

#### 4.8.1 Zmena GUI z nového vlákna

Ďalším problémom, ktorý sa týka priamo zdrojového kódu aplikácie bola nemožnosť meniť grafické prvky GUI z iného vlákna ako z hlavného. Keďže niektoré funkcie programu bežia v samostatných vláknach tak bolo treba nájsť riešenie.

Pomocou dokumentácie a informácií, ktoré som našiel na odborných fórach, som sa dostal k takémuto riešeniu:

Každý príkaz na zmenu GUI ktorý je volaný z iného ako hlavného vlákna musí byť volaný pomocou metódy *Platform.runLater(Runnable r)*.

---

```
Platform.runLater(new Runnable(){
    @Override public void run() {
        // zmena GUI
    }
});
```

---

Táto metóda spúšťa nové vlákno na hlavnom aplikačnom vlákne, ktoré sa vykoná niekedy v budúcnosti. Môže sa volať z akéhokoľvek vlákna. Po volaní sa toto volanie zaradí do fronty a vlákno nezávisle na to pokračuje ďalej. Volania vo fronte sa vykonávajú v poradí, v akom boli uskutočňované.

#### 4.8.2 Obnovenie listu v Okne UserProfiles

Keďže komponenty na Platfome JavaFX sú navrhnuté tak, aby sa obnovovali sami, aj keď nie vždy je tomu tak, neobsahujú metódu pre takzvaný manuálny refresh. V okne *UserProfiles* sa vyskytla chyba, kedy bolo treba v liste zmeniť zvýraznenie aktívneho profilu. Táto zmena sa pôvodne udiala len počas zavretia a následného otvorenia okna.

Toto zvýraznenie je realizované pomocou *CellFactory* a jediná možnosť bola priradiť toto overenie aktívneho profilu a jeho následne označenie preťažením metódy *requestFocus()*, kde sa po zmene označenia aktuálne označená položka overí či nieje aktuálne zvolený profil. V prípade zvoleného profilu sa označí šedou farbou, v opačnom prípade sa vráti do pôvodného stavu. Následne bolo treba vytvoriť metódu na postupné označenie týchto položiek tak aby po vybraní aktívneho profilu zostal označený ten správny.

---

```
private void refresh(){
    for(int i = 0; i < userProfilesListView.getItems().size(); i++){
        userProfilesListView.getSelectionModel().select(i);
    }
}
```

---

Pomocou tejto metódy si postupne poznačuje každá položka listu a tým sa zmení označenie na aktívne. Táto metóda sa volá zakaždým, keď je zmenený aktívny profil.

## 5 Záver

### 5.1 Uplatnené znalosti získané počas štúdia

Znalosti, ktoré som získal počas štúdia a mohol uplatniť počas praxe sú hlavne z predmetov ako *Programovacie jazyky I.*, *Java Technológie* kde som získal znalosti o programovacích jazykoch a hlavne o jazyku Java. Tieto znalosti mi boli veľmi užitočné pri implementácii nového grafického rozhrania, ako aj pri študovaní pre mňa novej platformy Java FX.

Ako ďalší predmet ktorý mi bol počas praxe užitočný by som spomenul *Spojovací systavy*. Znalosti z tohto predmetu som taktiež využil, hlavne znalosti o protokole SIP, ktoré som potreboval.

### 5.2 Chýbajúce znalosti

Novinkami, s ktorými som sa ešte nestretol bola hlavne platforma Java FX. Taktiež aj chýbajúce skúsenosti z prácou na väčších projektoch.

### 5.3 Zhodnotenie praxe

Prax vo firme Siemens EN považujem za veľmi prínosnú skúsenosť. Som rád že som mohol pracovať v tejto firme a získať tak neoceniteľné skúsenosti čo sa týka práce na väčších projektoch, ako aj osvojenie si technológie Java FX.

Na projekte, ktorý mi bol pridelený zostávam pracovať aj po skončení praxe, pretože bol rozsiahlejší a na jeho dokončenie je potrebné väčšie množstvo času ako som mal počas mojej odbornej praxe.

## 6 Literatúra

- [1] Siemens: Company Information. SIEMENS ENTERPRISE COMMUNICATION. *Siemens-enterprise.com* [online]. 2011 [cit. 2013-04-28].  
Dostupné z: <http://www.siemens-enterprise.com/sk/about/company-information.aspx>
- [2] NetBeans IDE Features. NETBEANS. *Netbeans.org* [online]. 2013 [cit. 2013-04-26].  
Dostupné z: <https://netbeans.org/features/index.html>
- [3] JavaFX Tools: JavaFX Scene Builder. ORACLE. *Oracle.com* [online]. 2013, 18.4.2013 [cit. 2013-04-22].  
Dostupné z: <http://www.oracle.com/technetwork/java/javafx/tools/index.html>
- [4] Oracle: Learn About Java Technology. ORACLE. *Java.com* [online]. 2010 [cit. 2013-04-26].  
Dostupné z: <http://www.java.com/en/about/>
- [5] Oracle: JavaFX - The Rich Client Platform. *Oracle.com* [online]. 2013, 16.4.2013 [cit. 2013-04-26].  
Dostupné z: <http://www.oracle.com/technetwork/java/javafx/overview/index.html>
- [6] SIP: Session Initiation Protocol. *Ietf.org* [online]. 2002 [cit. 2013-04-27].  
Dostupné z: <http://tools.ietf.org/html/rfc3261>
- [7] SIPp: Welcome to SIPp. *Sourceforge.net* [online]. 2013 [cit. 2013-04-27].  
Dostupné z: <http://sipp.sourceforge.net/>